Institute for Advanced Simulation

# Coarse Grained Electronic Structure Using Neural Networks

Jörg Behler

published in

*Multiscale Simulation Methods in Molecular Sciences*,
J. Grotendorst, N. Attig, S. Blügel, D. Marx (Eds.),
Institute for Advanced Simulation, Forschungszentrum Jülich,
NIC Series, Vol. **42**, ISBN 978-3-9810843-8-2, pp. 247-270, 2009.



http://www.fz-juelich.de/nic-series/volume42

# Coarse Grained Electronic Structure Using Neural Networks

**Jörg Behler**

Lehrstuhl für Theoretische Chemie
Fakultät für Chemie und Biochemie
Ruhr-Universität Bochum, 44780 Bochum, Germany
*E-mail: joerg.behler@theochem.rub.de*

The accuracy of the results obtained in theoretical simulations critically depends on the reliability of the employed interatomic potentials. While efficient electronic structure methods like density functional theory (DFT) have found a wide application in molecular dynamics simulations of comparably small systems containing up to a few hundred atoms, for an investigation of many interesting questions one has to deal with systems too large for DFT. In recent years artificial neural networks (NN) have become a promising new tool for the representation of potential-energy surfaces (PES). Due to their flexibility they are able to accurately reproduce a given set of electronic structure data, while the resulting continuous NN-PES can be evaluated several orders of magnitude faster than the underlying electronic structure calculations. Additionally, analytic energy gradients are readily available making NN potentials well suitable for applications to large-scale molecular dynamics simulations. The main drawback of NN potentials is their intrinsically non-physical functional form. Consequently, large reference data sets from electronic structure calculations have to be available to construct reliable NN potentials, which are thus more costly to construct than conventional empirical potentials.

## 1 Introduction

The investigation of many interesting chemical problems requires long simulations of large systems containing hundreds or thousands of atoms. While in principle accurate electronic structure methods are available[1,2], a direct combination with molecular dynamics (MD) simulations "on-the-fly" is feasible only for small systems, and an application of these methods to large systems is in most cases prohibitively expensive. This dilemma is often circumvented by employing a so-called "divide and conquer" approach. In this approach the costly evaluation of accurate energies and forces by sophisticated electronic structure methods is separated from the actual simulation by a three step procedure. First, the potential is evaluated for a set of representative atomic configurations by highly accurate methods (sometimes also experimental data is used). In the second step a continuous potential representation is constructed, which can be evaluated much faster but should ideally provide essentially the same description as the underlying electronic structure methods. This potential then provides fast access to the potential-energy surface (PES). Finally, in the third step the simulations are carried out employing this potential, which typically allows an extension of length and/or time scales by many orders of magnitude. The use of this approach is wide-spread, a well-familiar example is the use of classical force fields[3–7] in MD simulations, and countless other empirical potentials of varying form and complexity have been developed in the past years for many types of systems.

Fitting complex potential-energy surfaces is a highly non-trivial task. The functional form has to be sufficiently flexible to adapt to the reference points with high accuracy, the

obtained PES should have continuous derivatives for applications in molecular dynamics simulations, it should be fast to evaluate, and its construction should not require a significant amount of manual work. Finally, an improvement and extension in certain regions of the configuration space should be possible without much effort, i.e., without starting the whole elaborate construction of a functional form and the fitting process right from the beginning, if new data points become available. The ideal method also would not be constrained to be applicable only to a certain type of system, like to certain classes of molecules or solids.

For simple systems analytical functional forms containing a few parameters can be "guessed", if possible based on physical knowledge about the system, and the parameters can be fitted to theoretical and/or experimental data. The choice of the functional form requires great care. If the functional form is not chosen appropriately, either unphysical artefacts may be introduced, or the shape of the PES is too much constrained and no accurate fit can be obtained. A well-known example for an analytic fit is the Lennard-Jones 12-6 potential[8]

$$V(r) = 4\epsilon \left\{ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right\} \tag{1}$$

with two parameters $\epsilon$ and $\sigma$. This functional form can well represent the interaction between two noble gas atoms, but there is no hope to describe complex organic molecules with this type of pair potential without any angular dependence. More elaborate potentials have been developed for many different types of systems. Classical force fields[3–7] are frequently used for large organic molecules, in particular for studies in the field of biochemistry. Their main drawback, the inability to describe the breaking and making of chemical bonds has been overcome for some systems by so-called "reactive force fields"[9–13]. Many other types of analytic potentials have been developed in recent years, and discussing even a small fraction is beyond the scope of the present lecture.

The general advantage of analytical fits is that the number of parameters is rather small, consequently only a few reference calculations are sufficient for setting up the potentials. Additionally, the individual terms often allow for a physical interpretation and an unexpected behavior of the potentials during simulations is rare. Problems may arise, if the functional form is too simple and thus cannot reproduce the reference data. For many systems the construction of suitable functional forms has been a frustrating challenge, and the sheer complexity of chemical bonding enforces either a limitation to a subset of possible structures that can be accurately described (e.g. by prohibiting bond breaking) or the chemical complexity has to be reduced. Most "reactive" potentials developed so far are thus applicable only to monocomponent or binary systems and have been mainly applied in the field of materials science[14–17] or to describe low dimensional PESs in surface science[18–20].

Another possibility to construct PESs is to use purely mathematical fitting methods like splines. They have a very general functional form, but they are not applicable to high-dimensional PESs, because the error increases rapidly with the number of dimensions, and they are very sensitive to noise in the data. An example for a very general approach for the description of molecule-surface interactions is the modified Shephard method[21,22], which is based on a Taylor expansion of the energy around the reference points.

In recent years, another mathematical approach based on artificial neural networks

(NN) has become a promising new tool for the construction of potential-energy surfaces. Artificial neural networks have been first introduced in 1943 by McCulloch and Pitts inspired by the way of signal processing in the brain[23]. The robustness, fault tolerance, ability to handle noisy data and highly parallel structure of the brain since then stimulated a lot of research and many attempts have been made to mimic these properties in computer algorithms. Nowadays, neural networks have found numerous applications in many fields of science. They have the general ability to associate some input pattern with a response, and are very well suited for applications like finger print identification, voice recognition and many others[24]. Apart from pattern recognition applications they can also be used to approximate functions based on a known set of function values[25].

There are various examples for applications in chemistry[26–29]: establishing structure-activity relationships[30], prediction of reaction probabilities[31], medical classification problems in clinical chemistry[32], binding site prediction[33], extraction of pair potentials from diffraction data[34], estimation of force constants in large molecules[35], "data mining" in drug design[36], electrostatic potential analysis[30,29], construction of exchange correlation potentials[37], the numerical solution of the Schrödinger equation for simple model systems[38], the prediction of secondary protein structure[39], and the detection of signals in the presence of noise[40].

The topic of this lecture is the application of neural networks to the construction of potential-energy surfaces employing their ability to approximate unknown functions very accurately. The central assumption behind the construction of potential-energy surfaces using neural networks is that an analytic representation of the PES exists. This analytic form may be very complex (indeed too complex to be ever written down) and completely unknown. But if it in principle exists, the neural network can be used to approximate this unknown functional form to high accuracy, since it has been proven that any real-valued function depending on a set of variables can be represented by a feed-forward neural network with arbitrary precision[25,41,42]. The reason for this capability of NNs is the extreme flexibility arising from a large number of simple non-linear functions which are combined in a hierarchical and systematic way. This is similar to the interconnection of biological neurons in the nervous system and gave the method its name. The structure of NNs is described in detail in the following sections.

Translating this capability of NNs to real world applications is not straightforward and a lot of effort has been put into the construction of neural network potentials by many research groups in chemistry and physics. The aim of this lecture is not to give a complete review of all techniques developed so far, but to point out some conceptual problems and how they can be solved for different types of chemical applications. For this purpose we will in particular focus on the class of multilayer feed-forward neural networks, which is most important for the representation of potential-energy surfaces.

## 2 Neural Network Potentials

### 2.1 The Functional Form of a Feed-Forward Neural Network

Since the introduction of artificial neural networks many different NN types have been developed[24,43–45]. For the representation of potential-energy surfaces in particular feed-forward neural networks have gained a central role. They have a very general, i.e. unbiased, form, which is a nested function of a large number of simple functional units. The

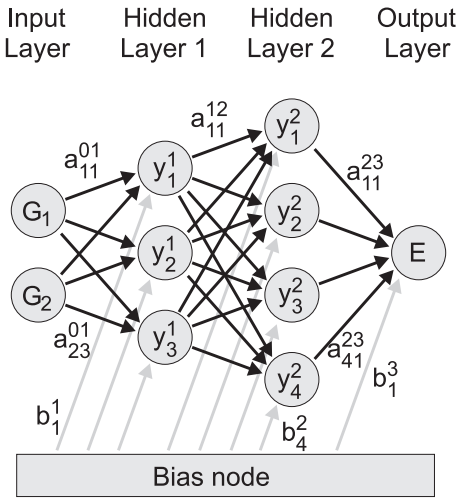


Figure 1. Schematic structure of a simple 2-3-4-1 feed-forward neural network. The value of the node in the output layer corresponds to the energy $E$, which depends on the variables $G_i$ in the input nodes. The bias weights $b_i^j$ connect the bias node with the nodes $i$ in layer $j$ and shift the non-linear region of the activation functions. The number of hidden layers and the nodes in these layers define the functional form of the NN. The weights parameters $a_{ij}^{kl}$ connecting node $i$ in layer $k$ with node $j$ in layer $l$ as well as the bias weights are the fitting parameters of the NN. They are determined iteratively using a set of known reference data points.

functional form of such a NN can be visualized schematically as shown for a small example NN in Fig. 1. The NN consists of artificial neurons called nodes and are represented by the grey circles. These nodes correspond to the neurons in the biological NN. They are arranged in several layers: an input layer, an output layer, and one or more hidden layers. The nodes in the input layer correspond to the $i$ variables $G_i$ of the potential-energy function, i.e., to the atomic degrees of freedom that determine the total energy of the system. The node in the output layer is the target quantity, the potential-energy of the system. The purpose of the remaining parts of the NN is to set up a functional relation between the atomic positions and the energy. The specific functional form is defined by the number of hidden layers and the number of nodes in each layer. The term "hidden layer" indicates that the nodes in these layers have no physical meaning and are just auxiliary mathematical constructs to provide the required flexibility. Each node in each layer is connected to the nodes in the adjacent layers by so-called weight parameters, the fitting parameters of the NN. Here, $a_{ij}^{kl}$ is the weight parameter connecting node $i$ in layer $k$ with node $j$ in layer $l$. The weight parameters are represented by the arrows in Fig. 1. Additionally, the nodes in the hidden layers and the output node are connected to a bias node via bias weight parameters. We use the symbol $b_i^j$, which is the bias weight acting on node $i$ in layer $j$.

Once the topology of the NN is set up, the output is calculated in the following way: First, the numerical values of the coordinates of the atoms in a given structure are provided to the NN in the nodes of the input layer, which has the layer index "0". The value $G_i$ of each node $i$ is then propagated to each node in the first hidden layer and multiplied by the value of the connecting weight parameter. At each node $j$ in the first hidden layer the sum of products $\sum_i G_i a_{ij}^{01}$ is calculated. So far this corresponds to a linear combination of

the atomic coordinates. Clearly, in the true PES there is a complicated non-linear relation between the energy and the atomic positions, and the capability to represent these general non-linear functions is introduced by applying a non-linear activation function $f_j^k$ to the final sum at each node $j$ in layer $k$. Examples for activation functions will be given below. Frequently used activation functions have a sigmoidal shape, i.e., they have a finite non-linear region and saturate for very small and very large arguments. The role of the bias weights $b_i^j$ is to shift the sum at each node into the non-linear regime of the activation functions (cf. Sec. 2.2). In summary, the value $y_j^1$ of node $j$ in the first hidden layer is then calculated by

$$y_j^1 = f_j^1 \left( \sum_i b_j^1 + a_{ij}^{01} G_i \right) \tag{2}$$

and for a general node $j$ in layer $k$ the equation becomes

$$y_j^k = f_j^k \left( \sum_i b_j^k + a_{ij}^{k-1,k} y_i^{k-1} \right) \quad . \tag{3}$$

The number obtained at each node in the first hidden layer is then propagated to each node in the second hidden layer and again multiplied by the respective connection weight. At each node in the target layer again an activation function is applied and so forth until finally a number is obtained at the node in the output layer of the NN.

The full functional form of the example NN in Fig. 1 is given accordingly by

$$E = y_1^3 = f_1^3 \left( b_1^3 + \sum_{k=1}^{4} a_{k1}^{23} \cdot f_k^2 \left( b_k^2 + \sum_{j=1}^{3} a_{jk}^{12} \cdot f_j^1 \left( b_j^1 + \sum_{i=1}^{2} a_{ij}^{01} \cdot G_i \right) \right) \right) \tag{4}$$

In general the NN output depends on the topology of the NN, i.e., the number of layers and nodes per layer, the type of activation functions, and most importantly, the numerical values of the weight parameters. Initially, the weight parameters are chosen randomly and the output of the NN is of course very different from the correct potential-energy of the structure. But if for a set of reference structures the potential-energy is known, e.g. from electronic structure calculations, then an error function can be defined as the difference between the output of the NN and the known correct energy. This error function can then be minimized by optimizing the weight parameters until all example points are accurately reproduced by the NN. Details on the weight optimization are given in Section 3 below. This optimum set of weight parameters is then kept fixed and can be used to predict the energies of new (similar) structures not included in the reference set, for instance structures visited in the trajectory of a MD simulation.

The size of the NN is determined empirically for a given system by constructing fits with different numbers of hidden layers and nodes per layer, and choosing the structure which provides the most accurate fit. Care has to be taken in that large networks may contain too many parameters. The resulting high flexibility may yield overfitting, which has to be checked carefully as will be discussed in Sec. 3.4. As a general rule, if two NN architectures provide the same accuracy the one with less parameters should be preferred. It has also been suggested to adapt the number of nodes during the fit, e.g. by employing genetic algorithms[46], but due to the increased computational costs of this approach so far it did not find regular use in the context of potential-energy surfaces.

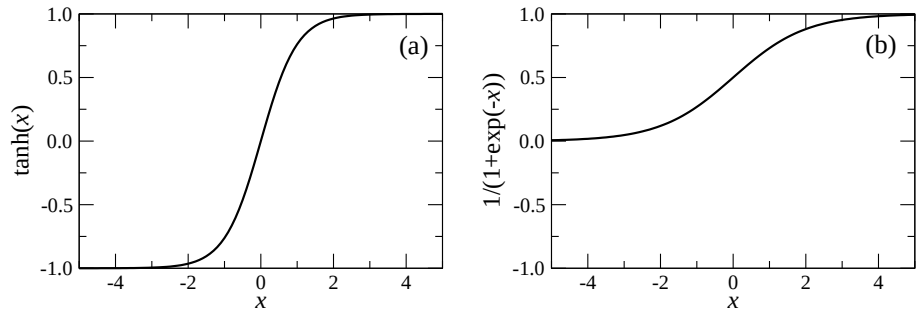


Figure 2. Frequently used activation functions: hyperbolic tangent (a), and sigmoid function (b). Activation functions saturate for very small and very large arguments, but have a non-linear region in between, which ensures that the neural network is able to adapt to general non-linear functions.

The topology of a feed-forward neural network can be described by a set of numbers defining the number of nodes in the input layer, each hidden layer and the output layer[47]. The simple example network in Fig. 1 in this scheme is a 2-3-4-1 NN. Additional letters can be provided describing the type of activation functions used in the individual layers, e.g. $t$ for a hyperbolic tangent, $s$ for a sigmoid function, and $l$ for a linear function (cf. Section 2.2).

### 2.2 Activation Functions

Neural networks obtain the ability to fit general, i.e., non-linear, functions by the incorporation of so-called activation functions. Activation functions are also called "transfer functions" or "basis functions" of the network. In general they map a variable $x$ to a range between -1 and 1 or between 0 and 1. This is a consequence of their general property that they saturate to these numbers for very small and very large values of $x$ and have a non-linear region in between. Frequently used examples for activation functions are the sigmoid function

$$f(x) = \frac{1}{1+e^{-x}} \quad , \tag{5}$$

the Gaussian function

$$f(x) = e^{-\alpha x^2} \quad , \tag{6}$$

or the hyperbolic tangent

$$f(x) = tanh(x) \quad . \tag{7}$$

The sigmoid function and the hyperbolic tangent activation functions are plotted in Fig. 2. For the output node sometimes also a linear function is used to avoid any constraint on the possible range of output values,

$$f(x) = x \quad . \tag{8}$$

Alternatively, if e.g. a hyperbolic tangent is applied, the range of energy values can be rescaled before the fitting to the interval between -1 and 1 that corresponds to the range of

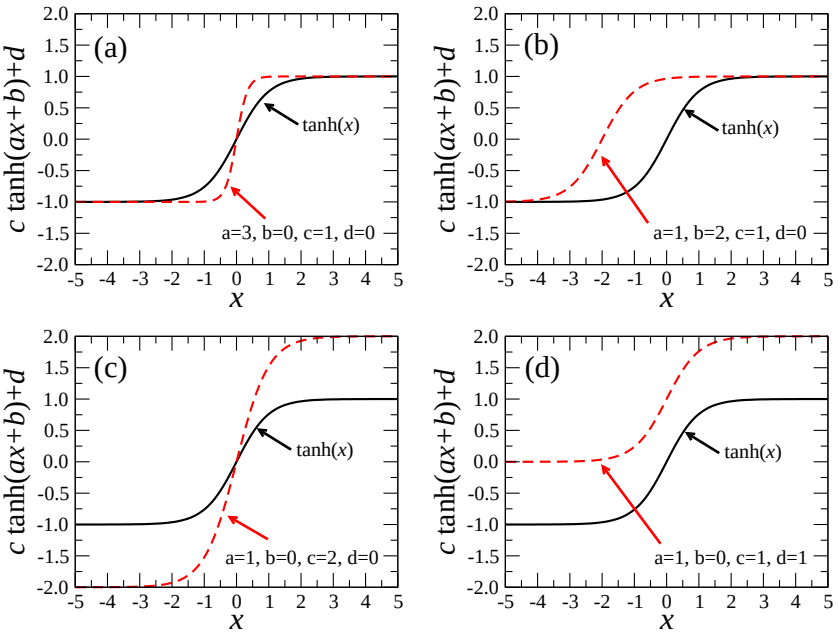


Figure 3. Illustration of the flexibility of the hyperbolic tangent activation function. The functional form of a neural network (Eq. 4) contains building blocks $f(x) = d + c \cdot \tanh(a \cdot x + b)$ which can adapt to general functions by varying the parameters $a$, $b$, $c$, and $d$. For comparison also the unmodified hyperbolic tangent is plotted as black line.

values of the hyperbolic tangent, and the values at the output node are scaled back to the original range. The activation functions have to be continuous and differentiable, which is needed for the application of standard optimization algorithms, but also for the calculation of the derivatives of the output with respect to the atomic coordinates, i.e., for the atomic forces.

We will illustrate the capability of the non-linear activation functions to adapt to arbitrary functions using the example of the potential of the harmonic oscillator

$$V(x) = x^2 \quad . \tag{9}$$

For instance for the hyperbolic tangent activation function the nested form of the neural network energy expression (e.g. Eq. 4) can be decomposed into a set of functional units of the form

$$f(x) = d + c \cdot \tanh(a \cdot x + b) \tag{10}$$

with four "parameters" $a$, $b$, $c$, and $d$. By optimizing these parameters, the shape of the hyperbolic tangent can be modified as illustrated in Fig. 3 This flexibility can be used to obtain a rather good approximation to the parabolic potential in a given range by just 2 activation functions, as shown in Fig. 4. Finally, we note that it has also been suggested to employ periodic activation functions, which can facilitate fitting periodic functions like torsional potentials[48].

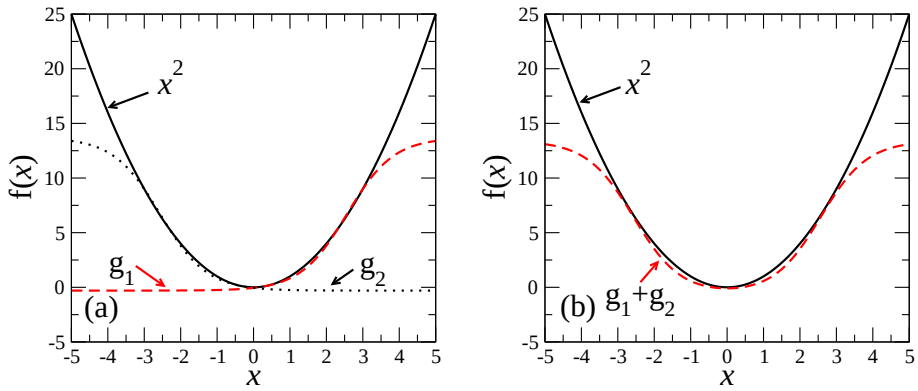


Figure 4. Example for the fit of a parabola in the range $-3 < x < 3$ by two hyperbolic tangent functions $g_1$ and $g_2$. In (a) the two functions are plotted separately, in (b) the sum is compared with the parabola.

### 2.3 Symmetry Functions

For several reasons it is advantageous not to use directly the Cartesian coordinates of the atoms as input for the neural network but to perform a transformation to coordinates which are physically more appropriate. These new sets of coordinates are often called "symmetry functions"[49]. This transformation is necessary because the numerical values of the Cartesian coordinates of the atoms do not carry the structural information defining the energy of the system in a directly usable form, but the relevant information is included in the relative positions of the atoms with respect to each other. A mere translation or rotation of the complete system must not change its energy, but it clearly affects the numerical values of the Cartesian coordinates. If these coordinates were used as input for the NN, the NN output would change with translation or rotation. A very basic coordinate transformation to avoid this problem is to switch from Cartesian to internal coordinates, i.e., defining the system in terms of bond lengths, bond angles and torsion angles. We note here that in contrast to classical force fields neural networks in general do not require the the specification of bonds ("bonded" atom pairs) and angles, so these terms could also be called two-, three- and four-body terms.

The use of internal coordinates provides a reasonable description of small molecular systems without any significant structural change like the breaking of bonds. Larger molecules, however, will contain many atoms of the same chemical species, and if two atoms of the same element are simply exchanged, the total energy of the system must not change. This symmetry information is not included in a set of internal coordinates, because the order in which the internal coordinates are fed into the NN is not arbitrary. Capturing this additional symmetry for general systems is a difficult task. For low-dimensional systems a sequence of symmetrization and antisymmetrization steps has been suggested[50,51]. To illustrate the procedure, imagine a water molecule with the bond lengths $r_{\mathrm{OH},1}$ and $r_{\mathrm{OH},2}$. A proper set of input coordinates for the NN has to take into account that both hydrogen atoms are indistinguishable, i.e., their numbering is arbitrary and exchanging the interatomic distances $r_{\mathrm{OH},1}$ and $r_{\mathrm{OH},2}$ must not change the output of the NN. By squaring the sum and the difference of $r_{\mathrm{OH},1}$ and $r_{\mathrm{OH},2}$ we obtain two functions $G_1$ and $G_2$, which

are independent of the numbering of the hydrogen atoms, i.e., exchanging both atoms does not change the values of $G_1$ and $G_2$.

$$G_1 = (r_{\mathrm{OH},1} + r_{\mathrm{OH},2})^2 \tag{11}$$

$$G_2 = (r_{\mathrm{OH},1} - r_{\mathrm{OH},2})^2 \tag{12}$$

In order to completely define this system with three degrees of freedom, a third coordinate for the distance between both hydrogen atoms or the angle $H_1$-O-$H_2$ can be introduced, which does not need to be symmetrized.

For an extension of this symmetrization/antisymmetrization scheme to high-dimensional systems soon numerical problems arise from the emergence of very large and very small numbers for the $G_i$ as well as from the increasingly complicated terms. This can be seen from the example methane, for which we give here the symmetrized terms for the four CH distances.

$$G_1 = \left[(r_{\mathrm{CH},1} + r_{\mathrm{CH},2})^2 + (r_{\mathrm{CH},3} + r_{\mathrm{CH},4})^2\right]^2 \tag{13}$$

$$G_2 = \left[(r_{\mathrm{CH},1} + r_{\mathrm{CH},2})^2 - (r_{\mathrm{CH},3} + r_{\mathrm{CH},4})^2\right]^2 \tag{14}$$

$$G_3 = \left[(r_{\mathrm{CH},1} - r_{\mathrm{CH},2})^2 + (r_{\mathrm{CH},3} - r_{\mathrm{CH},4})^2\right]^2 \tag{15}$$

$$G_4 = \left[(r_{\mathrm{CH},1} - r_{\mathrm{CH},2})^2 - (r_{\mathrm{CH},3} - r_{\mathrm{CH},4})^2\right]^2 \tag{16}$$

Many more symmetrized terms for the H-H distances and/or HCH angles are required to describe all degrees of freedom. It is immediately obvious that this scheme cannot be extended to larger molecules containing many atoms of the same species. Nevertheless, for low-dimensional potential-energy surfaces this scheme is very useful and can also be applied to molecule-surface interactions. In the latter case further compilations are related to the lateral periodicity of crystalline surfaces. For the construction of symmetry functions in this case the bond lengths have to be replaces by Fourier terms reflecting the surface symmetry[51].

In general, the choice of symmetry functions is system-dependent, but they have to fulfill several requirements. They need to be continuous in value and slope, they should be fast to evaluate, and there should be a one-to-one correspondence between a given structure and its set of symmetry function values. If two structures with different energies yield the same set of symmetry function values, fitting the NN is not possible, because the NN would associate two different energies to the same structure. It should also be noted that there is no need to ever invert the transformation of the coordinates. The mapping is always from atomic configurations to symmetry functions, for the construction of the training set as well as for the energy prediction of a new structure.

Typically the range of values of each symmetry function is scaled. This has numerical reasons, because it is advantageous to avoid symmetry function values in the saturation region of the activation functions. In this case the function values of the activation functions would be about the same for all symmetry function values. Further, depending on the definition, symmetry functions may have a very large or very small range of values, in particular if symmetrization/antisymmetrization is applied. Also in this case it is advantageous to rescale the range of values to an interval between 0 and 1.

In summary, it is usually not possible to use the Cartesian coordinates of the atoms to construct a NN PES. Instead, in a first step, the Cartesian coordinates are mapped onto a

set of symmetry functions $\{G_i\}$. In the next step the symmetry function values are used as input for the NN, which then yields the energy of a structure.

### 2.4 Atomic Forces

The analytic form of the neural network total energy expression in Eq. 4 allows to calculate analytic derivatives, which are essential to obtain accurate forces for applications in MD simulations. If an intermediate mapping of the atomic coordinates onto symmetry functions $\{G_i\}$ is used, the force with respect to an atomic coordinate $\alpha$ is given by

$$\begin{aligned} F_\alpha &= -\frac{\partial E}{\partial \alpha} \\ &= -\sum_i \frac{\partial E}{\partial G_i} \frac{\partial G_i}{\partial \alpha} \end{aligned} \tag{17}$$

The derivative $\frac{\partial E}{\partial G_i}$ is given by the NN topology, the derivative $\frac{\partial G_i}{\partial \alpha}$ is defined by the choice of symmetry functions. Also other quantities containing analytic derivatives like the stress tensor are directly accessible.

## 3 Optimization of the Weight Parameters

### 3.1 The Fitting Procedure

In order to predict the potential-energy of an atomic configuration, the weight parameters of the NN have to be known. Typically, these parameters are optimized iteratively using a set of known function values. This optimization process is called "training" or "learning" of the NN.

A large variety of algorithms can be used to optimize the weight parameters[52], which can be classified as gradient-based algorithms and random methods. Examples for gradient-based methods are the steepest-descent algorithm, which is called "back-propagation" in the NN community, conjugate gradients[53–55], the global extended Kalman filter[56], and many more. Gradient-based learning schemes are likely to get trapped in local minima at some point, but they are fast. Examples for random methods are the weight optimization employing genetic algorithms[52] or a swarm search[52]. Random methods can easily jump from one local minimum to another, but they are computationally very demanding. A method combining ideas from gradient-based and random methods is simulated annealing[57,35,52], which is essentially a damped dynamics in the space of the weight parameters.

For complex potential-energy surfaces and large data sets, which can easily reach the order of 10000 reference points, typical NNs contain between one and three hidden layers and between 25 and 40 nodes per layer. Consequently, roughly 1000 to 5000 weight parameters are used. The optimization of such a large number of weight parameters is a formidable task and there is no hope in practical fits to find the global minimum. Still, many local minima may represent the training set sufficiently well and can provide a reliable NN potential, and often many fits of a comparable quality are found with different sets of weight parameters. The resulting NNs, which yield about the same fit quality but cannot be transferred into each other by a simple permutation of the NN nodes, are called degenerate NNs[58].

The optimization of the weight parameters corresponds to the minimization of a cost function $\Gamma$, which is defined as the sum of the squared errors of the energies $E_{i,\mathrm{NN}}$ predicted by the NN and the "true" reference energies $E_{i,\mathrm{Ref}}$ from electronic structure calculations.

$$\Gamma = \sum_{i=1}^{N} \frac{1}{N} (E_{i,\mathrm{NN}} - E_{i,\mathrm{Ref}})^2 \tag{18}$$

It is also possible to modify this cost function by assigning different fitting weights (not to be confused with the weight parameters) to enforce a more accurate fit for certain parts of the PES, e.g. along the reaction path[49].

The optimization process is started by initializing the weight parameters as random numbers, typically in the range $[-1, 1]$. In each iteration, which is also called "epoch" in the NN community, each training point is presented once to the NN. Usually the training points are presented in random order, to reduce the probability of ending up in a close local minimum.

The measure for the quality of a fit is the root mean squared error of the training points

$$RMSE = \sqrt{\frac{1}{N} \sum_{i} (E_{i,\mathrm{NN}} - E_{i,\mathrm{Ref}})^2} \tag{19}$$

which is calculated in every epoch. Sometimes also the mean absolute deviation

$$MAD = \frac{1}{N} \sum_{i} |E_{i,\mathrm{NN}} - E_{i,\mathrm{Ref}}| \tag{20}$$

is monitored. The course of the RMSE in a typical fit will be discussed in Section 3.4.

Because of their efficiency compared to random fitting methods, gradient-based algorithms play a dominant role. For their application the partial derivatives of the NN output

$$\frac{\partial \Gamma}{\partial a_{ij}^{kl}} = \sum_{\mu} \frac{1}{\mu} \frac{\partial E_{\mu,\mathrm{NN}}}{\partial a_{ij}^{kl}} \tag{21}$$

and

$$\frac{\partial \Gamma}{\partial b_{i}^{j}} = \sum_{\mu} \frac{1}{\mu} \frac{\partial E_{\mu,\mathrm{NN}}}{\partial b_{i}^{j}} \tag{22}$$

with respect to all weight parameters have to be calculated for each training point. There are two types of learning. In the so-called "offline" learning the weights are updated once per epoch, e.g. if a conjugate gradient is applied. In "online" learning the weights are updated after the presentation of each individual training point. In this case the summation in Eqns. 21 and 22 are dropped and the gradients for each single point are used separately. An example for an algorithm for online learning is the global extended Kalman filter. In the following Sections two frequently used optimization algorithms, back-propagation and the global extended Kalman filter will be discussed.

### 3.2 Back-Propagation

The most frequently employed algorithm for the optimization of the weight parameters is "back-propagation". Essentially, this is identical to a standard steepest-descent optimization. In the back-propagation optimization the weight parameters are updated according to

$$a_{ij,new}^{kl} = a_{ij,old}^{kl} - \kappa \frac{\partial \Gamma}{\partial a_{ij}^{kl}} \tag{23}$$

$\kappa$ is a positive damping factor called "learning rate". The term "back-propagation" has its origin in the way the derivatives with respect to the weights are calculated. The output of a feed-forward NN is calculated in a so-called "forward-pass" through the NN. First, using the values at the input nodes and of the connecting weights the values at the nodes in the first hidden layer are determined. They are then passed forward to the second hidden layer, to evaluate the numerical values at the nodes in this layer and so on. Consequently, the total output of the NN is calculated by propagating the information forward through the NN. On the other hand, for the calculation of the derivatives of the NN output value with respect to the connecting weights the information flow is in the opposite direction, "backwards".

As a steepest-descent method back-propagation is not very efficient and likely to get trapped in a local minimum. It may also show an oscillating behavior or diverge if $\kappa$ is chosen too large, and the optimum value of $\kappa$ is system-dependent.

### 3.3 The Kalman Filter

An optimization scheme which has become very popular in the context of neural networks is the extended Kalman filter. The global extended Kalman filter is a very sophisticated algorithm originating from estimation and control theory[59]. It is used for online learning, i.e., the weight parameters are optimized after the presentation of each individual training point. In the Kalman filter the update is directed by a weighted history of previous updates of the weight parameters. The derivation of the equations used in the weight update is beyond the scope of the present lecture, and here we will just present the result for the Kalman filter recursion relations for the update $n$:

$$\mathbf{K}(n) = \lambda^{-1}\mathbf{P}(n-1)\mathbf{J}(n)\left[\mathbf{I} + \lambda^{-1}\mathbf{J}^T(n)\mathbf{P}(n-1)\mathbf{J}(n)\right]^{-1} \tag{24}$$

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{K}(n)\mathbf{J}^T(n)\mathbf{P}(n-1) \tag{25}$$

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{K}(n)\left[\mathbf{E}_{\mathrm{Ref}}(n) - \mathbf{E}_{\mathrm{NN}}(\mathbf{w}(n-1)\right] \tag{26}$$

$\mathbf{K}$ is the Kalman gain matrix, and $\mathbf{J}$ is the Jacobi matrix with the elements

$$J_i = \frac{\partial E}{\partial w_i} \tag{27}$$

where $w$ is either a connection or a bias weight parameter. $\mathbf{P}$ is the covariance matrix, and $\mathbf{I}$ is the identity matrix. For each training point first the Kalman gain matrix is updated using the covariance matrix of the previous epoch and the current weight derivatives in the Jacobi matrix. Then the new vector of weight parameters $\mathbf{w}$ is determined using $\mathbf{K}$. Finally, the covariance matrix is updated according to Eq. 25. A "forgetting schedule" is

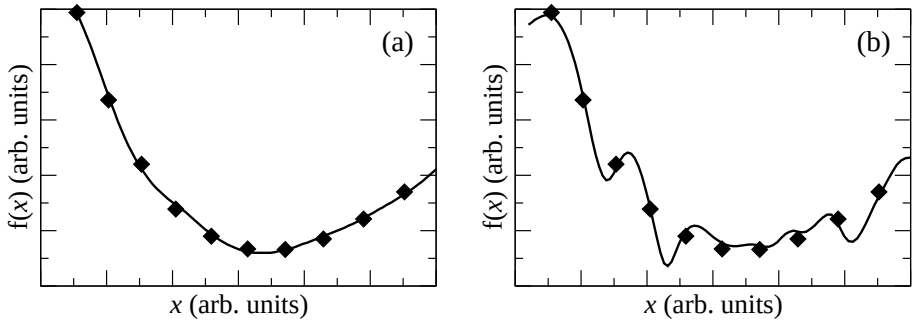


Figure 5. Illustration of "overfitting". In (a) the training points (diamonds) are well represented and smoothly connected by the fit (line). Also in (b) the fit is very accurate for the training points, but many local extrema are present in between the training points. These extrema are not based on information in the training set but are artefacts of the fit arising from too much flexibility. This "overfitting" can be detected for example by using a test set of point located in between the training points. If the error of the test points is significantly higher than the error of the training points, overfitting is present.

introduced via $\lambda$ to ensure that only the recent history of updates in taken into account for the update of point $n$,

$$\lambda(n) = \lambda_0 \lambda(n-1) + 1 - \lambda_0 \quad . \tag{28}$$

The constant $\lambda_0$ is usually chosen between 0.99000 and 0.99900.

Adapting the weight parameters after each training point is computationally rather costly. In its adaptive form the extended Kalman filter thus is not used to update the weight parameters after each individual training point, but an error threshold $\alpha$ is defined in terms of the actual RMSE of the full training set. Only if the error of a training point is larger than the product of $\alpha$ and the current RMSE, the point will be used to update the weights. This can reduce the computational effort significantly, since only points are used in the fit, which are not well represented.

For the construction of NN potentials, the extended Kalman filter often shows a performance which is superior to other optimization algorithms[56,58], because is less likely to get trapped in shallow local minima.

### 3.4 Overfitting

Employing a very flexible functional form immediately rises the question, how overfitting can be detected and controlled. If a set of training points is fitted very accurately while other points not included in the training set are poorly described, this is called "overfitting". In other words, overfitting is an improvement of the fit in one region of the configuration space at the cost of a poor quality in another region. This is illustrated in Fig. 5. In (a) the training points are well represented and connected by a smooth curve which seems to be a reasonable fit. The RMSE of the training points will be very low in this case. In (b), however, the RMSE will be very low as well, possibly lower than in (a), because also here the curve is very close to all the training points indicated as diamonds. Nevertheless, in (b) the curve shows many local extrema, which are apparently not justified by the training data. This is a typical example for overfitting.

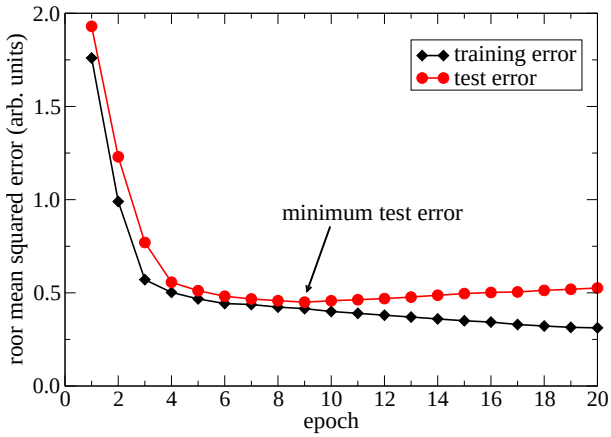


Figure 6. Typical course of the errors of the training set and the test set in the iterative optimization of the NN weight parameters.

Most applications of neural networks do not allow for a visual inspection of the fit quality due to the high dimensionality. A commonly employed method in these cases is the so-called "early stopping" method. In this method the available points are split into a training set which is used to update the weight parameters, and a test set, which is not used in the optimization procedure. A comparison of the RMSEs of the training and the test set then allows for an estimate of the generalization properties of the NN fit. The typical course of the training error and the test error in the iterative fit is shown in Fig. 6. The training error decreases steadily since the weight parameters are optimized to reproduce the training set as accurately as possible. Initially, also the test error drops quickly, because the description of the overall shape of the potential-energy surface is improving in each epoch. Then the test error reaches an local minimum and starts to increase slowly. This increase indicates that now the accuracy of the training points is improved on the expense of the regions in configuration space in between the training points. This is detected by the RMSE of the test points, which are located in between the training points. In the "early stopping" method, the set of weights, which minimizes the error of the test set is considered to represent the best fit.

## 4 Construction of the Training Set

The training set for the optimization of the weight parameters can be obtained from any electronic structure method, because the only information required for each atomic configuration in the training set is its total energy. Information on energy gradients can in principle also be used in the construction of the NN potentials[58], but in practice this is rarely done. A significant constraint on the choice of the electronic structure method for large systems is the large number of training points that is needed to set up a neural network potential. This limits the application of computationally demanding but very accurate quantum chemical methods to small molecules, and the most frequently used electronic structure method for large systems is density functional theory.

Having chosen an electronic structure method for the calculation of the training set, the next problem is the choice of the atomic configurations. For small systems with few degrees of freedom, e.g. small molecules, a dense grid of points can be obtained by systematically varying all degrees of freedom. However, the exponential growth of the number of configurations prevents a systematic mapping for larger systems of typically more than six degrees of freedom. This is because if each degree of freedom is sampled by $N$ points, for $d$ degrees of freedom the number reference calculations is $N^d$. In practical applications like MD simulations often only a subspace of the full configuration space is accessible for the system. This relevant subspace can be mapped by a systematic approach in the following way: First, some random structures are calculated and a preliminary NN potential is constructed using these points. This potential will not be reliable and may contain unphysical stationary points. Nevertheless it can be used to perform short MD simulations or structural relaxations to propagate the system to new configurations. The configurations suggested in this way by the NN can then be recalculated using electronic structure methods, and the resulting energies can be compared with the NN predictions. If the agreement is not satisfactory, the new structures can be included in the training set and the fit can be refined. In contrast to conventional empirical potentials with a given analytic form and a few adjustable parameters no change in the functional form of the NN is required, and improving the NN is straightforward without any manual work. The new fits again can be used to suggest new structures and so forth, until all wrong features of the NN PES have been removed and the training data set is reproduced with the desired accuracy. Typically, an accuracy of a few meV per atom with respect to the reference energies can be obtained in this way.

It is also possible, to identify regions of the configuration space, which are relevant but not well represented in the training set, without carrying out costly electronic structure calculations. For this purpose several NN fits are constructed employing different NN topologies. Because the NN topologies are different, so is the functional form of the fits. Now two fits with approximately the same RMSEs for the training and the test set are chosen. Accordingly, it is not possible to judge which of the two fits is a better representation of the true PES. Then a large number of structures is generated, e.g., random structures, optimized structures or snapshots from MD simulations. The energy for all structures is predicted by both fits. If a structure is close to a point already included in the training set, both NNs are likely to predict a similar energy, otherwise the RMSEs of the NNs would be clearly different. But if the predicted energies are very different, then the NNs have too much flexibility at this point in the configuration space and an electronic structure calculation should be carried out for this point. This way it is possible to avoid a large number of unnecessary electronic structure calculations.

## 5 Applications of Neural Network Potential-Energy Surfaces

### 5.1 Neural Network Potentials for Molecular Systems

To date, the most frequent application of NN potentials is the representation of rather low-dimensional molecular PESs. Many different sets of coordinates have been developed to transform the atomic positions to suitable inputs for the NN. For simple diatomic molecules the potential depends only on the interatomic distance and one input coordinate is sufficient

for the PES construction. An example is a study of the photodissociation of the $HCl^+$ ion using NN PESs for the ground and excited state[60]. Also transition probabilities in the OH radical have been studied[61]. However, for larger molecules the coordinates can become significantly more complicated. Ideally, the coordinate transformations takes the molecular symmetry directly into account. The exact form of the applied set of symmetry functions is highly system-specific. In a study of the vibrational levels of the $H_3^+$ ion a symmetrical formulation has been suggested taking into account the equivalence of all three nuclei[62,63]. NNs have also been used to describe the interaction between two HF molecules as well as between a HF and a HCl molecule[64].

NN potentials can be very useful in situations when the polarizability of molecules complicates the construction of classical potentials. This has been shown for the example of the $Al^{3+}$ ion in water by combining conventional two-body terms with three-body interactions, which are represented by a flexible NN[50]. The equivalence of the two water molecules interacting with the $Al^{3+}$ has been included by symmetrized coordinates. NN potentials have also been constructed for the water dimer[65], and later applied to liquid water in combination with empirical parameters of the TIP4P water model[66]. A very general method for molecules has been suggested based on a high-dimensional model representation employing a many-body expansion of the potential[67–70]. This approach is very systematic and promises a very high accuracy, but due to its complexity and computational demand, it is still limited to a rather small number of degrees of freedom in practical applications.

### 5.2 Neural Network Potentials for Molecule-Surface Interactions

Neural network potentials have also been applied to the description of molecule-surface interactions[47,49,71]. In contrast to molecular systems the number of degrees of freedom is typically much larger, because realistic surface models for instance in form of periodically repeated slabs contain a significant number of atoms. Two approaches have been followed in order to reduce the resulting complexity: either only a few degrees of freedom of selected surface atoms are included[47], or a frozen surface approximation is applied, i.e., all degrees of freedom of the surface atoms are eliminated by freezing their positions. For diatomic molecules this approach reduces the problem to a six-dimensional potential-energy surface, which can then be mapped systematically on a grid of a few thousand points by electronic structure calculations. The frozen surface approximation is a drastic approximation and its validity has to be checked carefully for each individual system. It has been found that physical quantities, which are less sensitive to a motion of the surface atoms, can be calculated to good accuracy[51,72–74,49].

Applications to molecule-surface interactions require a special type of symmetry functions[71,51], which have to include the periodicity of the surface as well as all symmetry elements of the surface unit cell. A transformation of the molecular coordinates to these symmetry functions is then equivalent to folding the configuration space into the symmetry unique wedge of the surface. An example of the symmetry unique wedge of the (111) surface of an fcc metal is shown in Fig. 7. This significantly reduces the computational costs for the calculation of the reference electronic structure calculations, because only configurations inside the symmetry unique wedge have to be calculated. Whenever the molecule leaves this wedge in the course of the trajectory, its coordinates are folded back

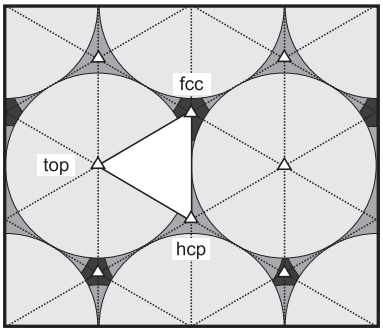


Figure 7. Symmetry unique wedge of the (111) surface of an fcc metal spanned by the top, fcc and hcp sites. The mirror planes perpendicular to the surface are indicated as dotted lines, the triangles represent threefold rotational axes.

to an energetically equivalent position in the symmetry unique wedge. In MD applications the NN PES needs to be continuous in value and slope. A special requirement for the description of molecule-surface interactions is that the symmetry functions have continuous derivatives at the boundaries of the symmetry unique wedge to avoid discontinuities of the atomic forces. A detailed discussion of a scheme for the systematic construction of suitable symmetry functions can be found elsewhere[51].

### 5.3 High-Dimensional NN Potentials for Condensed Systems

To date NN potentials have been mainly applied to PESs of gas phase molecules with a rather low number of up to about 12 degrees of freedom. For general chemistry in condensed systems, e.g. in solution, in the solid state or at surfaces, an extension of the NN methodology to high-dimensional PESs explicitly depending on hundreds of degrees of freedom is required. This cannot be achieved in a brute-force approach by simply increasing the number of input nodes, because a systematic mapping of the associated configuration space is too costly. Further, the efficiency of the NN evaluation decreases with increasing number of input nodes. Finally, the internal structure of the NN constitutes a major obstacle in that the input nodes of the NN are "ordered". Each node in the input layer is connected to all nodes in the first hidden layer, but the numerical values of the connecting weights are all different. For larger systems containing many atoms there is necessarily an invariance of the total energy with respect to the exchange of atoms of the same element. Any high-dimensional NN approach must take this invariance into account. Recently, several extensions of the NN methodology have been suggested to address systems with a large number of degrees of freedom.

An extension of NN PESs to in principle arbitrary dimensionality has been proposed by employing the NN to fit the many-body term of an empirical potential[75]. Specifically, the functional form of the Tersoff potential has been used[15]:

$$V_{\mathrm{Tersoff}} = \frac{1}{2}\sum_i \sum_{j\neq i} f_c(r_{ij})\left[V_R(r_{ij} - b_{ij}V_A(r_{ij}))\right] \tag{29}$$

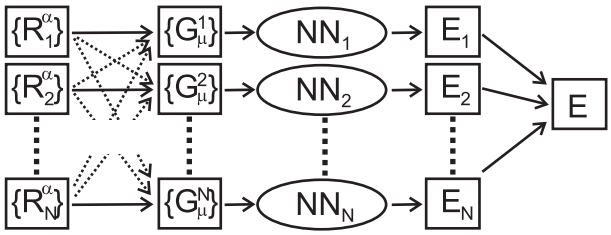


Figure 8. Schematic structure of a high-dimensional neural network potential for a condensed systems containing $N$ atoms[79]. Each atom contributes the energy $E_i$ to the total energy $E$. The local geometric environment of each atom with the Cartesian coordinates $\left\{R_i^\alpha\right\}$ is described by a set of symmetry functions $\left\{G_\mu^i\right\}$. The set of symmetry functions of each atom thus depends on the positions of all other atoms up to a cutoff radius as described by the dotted arrows. The symmetry functions are then used as input values for atomic neural networks (NN), which yield the $E_i$. The NN can be applied to any system size, because all atomic NNs have are identical, and for each additional atom one line has to be added to this scheme.

$V_R$ and $V_A$ are repulsive and attractive two-body terms. The attractive term is scaled by a many-body term $b_{ij}$, which is now replaced by a NN. The method has been applied to a binary system containing C and H[75] and to elemental silicon[76,77]. Its main drawbacks are the constrained functional form due to the combination with a rather simple empirical potential and the limitation to binary systems. Still, applications to large bulk structures and clusters with an explicit energy dependence on all atomic degrees of freedom have become possible showing promising results. Recently, this approach has been reformulated in terms of a general many-body expansion including two-, three- and four-body interactions[78].

Another approach for the construction of high-dimensional NN PESs avoiding any incorporation of empirical functional forms is based on a decomposition of the total energy $E$ of the system into atomic energy contributions $E_i$[79]. This ansatz, which is also used in many empirical potentials, is based on the assumption that the energy contribution of each atom $i$ in the system is determined by its local chemical environment up to a certain cutoff radius $R_c$. The total energy of the system is then constructed as a sum of the energy contributions of all atoms in the system

$$E = \sum_i E_i \tag{30}$$

The atomic energy contributions are calculated by atomic neural networks. The procedure is shown schematically in Fig. 8. Each atom in the system corresponds to one line in this scheme. First, the Cartesian coordinates $R_i = \{X_i, Y_i, Z_i\}$ of atom $i$ are transformed to a set of symmetry functions $\{G_i\}$, which describe the local environment of this atom. The $\{G_i\}$ thus can be regarded as a kind of structural fingerprint, which is then used as input for an atomic NN. Typically 40-50 symmetry functions are used for each atom. This corresponds to an effective reduction of the dimensionality of the problem, because for each atom only neighbors inside the cutoff sphere determine the symmetry function values as indicated by the dotted arrows in Fig. 8. The cutoff radius $R_c$ is typically chosen in the order of 6 Å, and the cutoff function $f_c$ is defined as

$$f_{\mathrm{c}}\left(R_{ij}\right) = \begin{cases} 0.5 \cdot \left[\cos\left(\frac{\pi R_{ij}}{R_{\mathrm{c}}}\right) + 1\right] & \text{for } R_{ij} \leq R_{\mathrm{c}} \\ 0 & \text{for } R_{ij} > R_{\mathrm{c}}, \end{cases} \tag{31}$$

with $R_{ij}$ being the distance between atom $i$ and its neighbor $j$. At the cutoff radius $f_c$ has zero value and slope. For the description of the local atomic environments many-body terms are used that depend explicitly on the positions of all atoms in the cutoff sphere. A "radial" symmetry function

$$G_i = \sum_j e^{-\eta_1 (R_{ij} - R_\mathrm{s})^2} \cdot f_\mathrm{c}\left(R_{ij}\right) \tag{32}$$

and an "angular" symmetry function for the angle $\theta = \frac{\mathbf{R_{ij}} \cdot \mathbf{R_{ik}}}{R_{ij} R_{ik}}$ centered at atom $i$, with $\mathbf{R_{ij}} = \mathbf{R_i} - \mathbf{R_j}$,

$$\begin{aligned} G_i = 2^{1-\zeta} \sum_\theta \left(1 + \gamma \cos\theta\right)^\zeta \cdot e^{-\eta_2 \left(R_{ij}^2 + R_{jk}^2 + R_{jk}^2\right)} \\ \cdot f_\mathrm{c}\left(R_{ij}\right) \cdot f_\mathrm{c}\left(R_{ik}\right) \cdot f_\mathrm{c}\left(R_{jk}\right) \end{aligned} \tag{33}$$

have been employed. These functions depend on parameters $\eta_1$, $\eta_2$, $\gamma$, $R_\mathrm{s}$ and $\zeta$, which define the region of configuration space described by the symmetry functions, and which are not optimized. A set of many different symmetry functions of these types is typically used differing in the parameter values. The radial functions can be interpreted as effective coordination numbers at various distances, the angular functions as angle distributions. Further details on the symmetry functions can be found elsewhere[79,80].

The outputs of the atomic NNs are the atomic energies $E_i$, which are finally added to yield the total energy of the system. This scheme has several advantages: first, the topology as well as the weight parameters of all atomic NNs are identical, thus exchanging two atoms in the system does not change the total energy. Secondly, once the weight parameters of the atomic NN have been determined, the NN PES can be applied to systems of varying size, because for each additional atom simply an atomic NN is added in the scheme in Fig. 8. Finally, because all symmetry functions are high-dimensional many-body functions, they are very well suited to describe systems with strong many-body interactions like metals. A drawback of this approach is related to the finite range of the atomic interactions, which is defined by the cutoff radius. If long-range interactions are important, which is typically the case for systems with charge transfer, the accuracy of the approach will be strongly reduced unless these interactions are explicitly taken into account. Thus, up to now only a few applications for elemental solids exist[81,80,82].

## 6 Discussion

In general, neural networks provide a very general and unbiased way to construct accurate potential-energy surfaces. However, there are also some drawbacks that need to be discussed. First of all, NNs provide analytic energy expressions which do not allow for a physical interpretation of individual terms. Internally, NNs remain a "black box", and the reliability of the PES has to be checked carefully.

A serious problem of NN potentials is related to the very flexible functional form, which is the reason for the accurate representation of the training points, but is *a priori* non-physical. Consequently, neural networks are very accurate for the energy prediction of structures similar to the structures included in the training set, but they can spectacularly fail for very different structures. An illustrative example is given in Fig. 9. It shows the

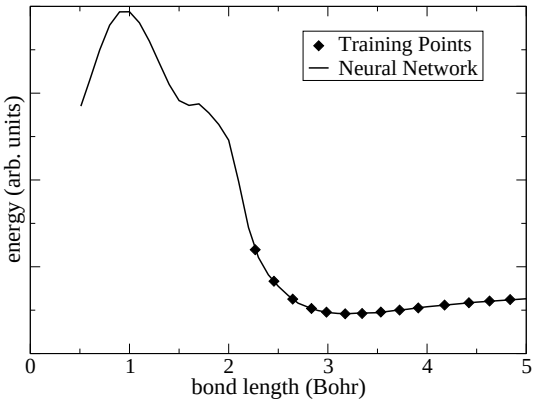


Figure 9. Demonstration of the poor extrapolation capabilities of neural networks (NN) for the binding curve of a dimer. In the range well represented by training points (solid diamonds) the NN potential (straight line) is very accurate, but because the NN does not include any assumptions on the functional form of the potential-energy surface, the highly repulsive part of the potential at short bond lengths is not described correctly. Instead the superposition of the tails of the activation functions give rise to an unphysical and unpredictable shape of the curve.

NN fit of a typical pair-potential of a dimer using a set of training points indicated as diamonds. In the range of bond lengths well-represented by the training points the NN energy curve is close to the reference data. For very short bonds however, the potential should be highly repulsive. Instead an obviously unphysical curve shape is obtained arising from a superposition of the tails of the activation functions. Because no training points are present in this range, this problem cannot be detected by an inspection of the RMSEs of the training set and the test set, which is typically distributed in the same range of values as the training set. However, it is straightforward to identify these regions by comparing for each input node of the NN the minimum and maximum values, which are present in the training set, with the symmetry functions values of a new atomic configuration. If the energy is requested for a structure whose symmetry function values are outside this range, the energy prediction should be taken with great care. Neural networks are designed for accurate interpolation, but they are not reliable in case of extrapolation. In molecular dynamics applications one has to make sure that the NN potential is "complete", i.e., there must not be any energetically accessible parts of the configuration space which are not well represented in the training set.

An advantage of NN potentials is that in contrast to classical force fields[3–7] they do not require a classification of atoms in terms of functional groups or hybridization state, and no bonds need to be specified. Neural network potentials are intrinsically "reactive" and in the course of an MD simulation based on a NN potential bonds can be broken or formed. Like in the underlying electronic structure calculations, just the atomic positions have to be provided in form of a suitable set of coordinates.

The use of system-specific symmetry functions allows to include information on the molecular symmetry in the NN, but often the construction of suitable symmetry functions is not straightforward. A combination of symmetry functions with a structural partitioning of the system into local environments as described in Section 5.3 now enables an application

to high-dimensional systems, but the exponential growth in the number of possible atomic configurations with the number of elements makes it unlikely for the near future that NN potentials will become a serious competitor for classical force fields e.g. in the field of biochemistry, unless their major advantages of "reactivity" is sacrificed by reducing the NN to fitting given parts of force fields. On the other hand, chemical problems involving a large number of atoms but a moderate number of elements, like in materials science, solid state chemistry, and some fields of surface science might strongly benefit from NN potentials in the future.

## 7 Concluding Remarks

In summary, the basic methodology and some technical aspects of the application of artificial neural networks to the construction of potential-energy surfaces for chemical reactions have been reviewed. Neural networks represent flexible functions, which are able to reproduce a given set of electronic structure data with high accuracy and to provide interpolated energies for similar structures. The resulting NN PES is computationally many orders of magnitude faster to evaluate than efficient electronic structure methods like density functional theory. The NN methodology itself is not bound to any particular reference total energy method and can also be applied in combination with wave-function based quantum chemical methods. The availability of analytic energy gradients enables a straightforward calculation of atomic forces for molecular dynamics simulations. The high flexibility, which is the reason for the numerical accuracy, is also the major drawback of NNs, namely the non-physical functional form. A large number of reference points is required to train a reliable NN PES and an extrapolation of the energy of structures very different from the structures included in the training set is not possible. The main future challenges are an extension of the applicability of NN potentials to high-dimensional multicomponent systems and the development of more systematic strategies for the construction and transferability checks of these potentials.

## References


1. A. Szabo, and N.S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover, 1996.
2. W. Koch, and M.C. Holthausen, *A Chemist's Guide to Density Functional Theory*, Wiley-VCH, 2001.
3. N.L. Allinger, Y.H. Yuh, and J.-H. Lii, J. Am. Chem. Soc. **111**, 8551 (1989).
4. S.L. Mayo, B.D. Olafson, and W.A. Goddard III, J. Phys. Chem. **94**, 8897 (1990).
5. A.K. Rappe, C.J. Casewit, K.S. Colwell, W.A. Goddard III, and W.M. Skiff, J. Am. Chem. Soc. **114**, 10024 (1992).
6. B.R. Brooks, R.E. Bruccoleri, B.D. Olafson, D.J. States, S. Swaminathan, and M. Karplus, J. Comp. Chem. **4**, 187 (1983).
7. W.D. Cornell, P. Cieplak, C.I. Bayly, I.R. Gould, Jr., K.M. Merz, D.M. Ferguson, D.C. Spellmeyer, T. Fox, J.W. Caldwell, and P.A. Kollman, J. Am. Chem. Soc. **117**, 5179 (1995).
8. J.E. Lennard-Jones, Proc. Phys. Soc. **43**, 461, 1931.

9. A.C.T. van Duin, A. Strachan, S. Stewman, Q. Zhang, X. Xu, and W.A. Goddard III, J. Phys. Chem. A **107**, 3803 (2003).
10. A. C. T. van Duin, S. Dasgupta, F. Lorant, W.A. Goddard III, J. Phys. Chem. A **105**, 9396 (2001).
11. A. Strachan, E.M. Kober, A.C.T. van Duin, J. Oxgaard, W.A. Goddard III, J. Chem. Phys. **122**, 054502 (2005).
12. K.D. Nielson, A.C.T. van Duin, J. Oxgaard, W.-Q. Deng, and W.A. Goddard III, J. Phys. Chem. A **109**, 493 (2005).
13. M.J. Buehler, A.C.T. van Duin, and W.A. Goddard III, Phys. Rev. Lett. **96**, 95505 (2006).
14. J. Tersoff, Phys. Rev. B **39**, 5566, 1989.
15. J. Tersoff, Phys. Rev. Lett. **56**, 632, 1986.
16. D.W. Brenner, and B.J. Garrison, Phys. Rev. B **34**, 1304, 1986.
17. F.H. Stillinger, and T.A.Weber, Phys. Rev. B **31**, 5262, 1985.
18. A. Groß, S. Wilk, and M.Scheffler, Phys. Rev. Lett. **75**, 2718, 1995.
19. G. Wiesenekker, G.J. Kroes, and E.J. Baerends J. Chem. Phys. **104**, 7344, 1996.
20. C.M. Wei, A. Groß, and M.Scheffler, Phys. Rev. B **57**, 15572, 1998.
21. C. Crespos, M.A. Collins, E. Pijper, and G.J. Kroes, Chem. Phys. Lett. **376**, 566, 2003.
22. C. Crespos, M.A. Collins, E. Pijper, and G.J. Kroes, J. Chem. Phys. **120**, 2392, 2004.
23. W. McCulloch, and W. Pitts, Bull. Math. Biophys. **5**, 115, 1943.
24. C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
25. K. Hornik, M. Stinchcombe, and H. White, Neural Networks **2**, 359, 1989.
26. J. Zupan, and J. Gasteiger, *Neural Networks for Chemists*, VCH, Weinheim, 1993.
27. W. Duch, G.H.F. Diercksen, Comp. Phys. Comm. **82**, 91, 1994.
28. B.G. Sumpter, C. Getino, and D.W. Noid, Ann. Rev. Phys. Chem. **45**, 439, 1994.
29. J. Zupan, and J. Gasteiger, Anal. Chim. Acta **248**, 1, 1991.
30. J. Gasteiger, and J. Zupan, Angew. Chem. **105**, 510, 1993.
31. R. M. Agrawal, A. N. A. Samadh, L. M. Raff, M. T. Hagan, S. T. Bukkapatnam, and R. Komanduri J. Chem. Phys. **123**, 224711, 2005.
32. G. Reibnegger, G. Weiss, G. Werner-Felmayer, G. Judmaier, and H. Wächter, Proc. Natl. Acad. Sci. USA **88**, 11426, 1991.
33. M. Keil, R.E. Exner, J. Brickmann, J. Comp. Chem. **25**, 779, 2004.
34. G. Toth, N. Kiraly, and A. Vrabecz, J. Chem. Phys. **123**, 174109, 2005.
35. T.H. Fischer, W.P. Petersen, and H.P. Lüthi, J. Comp. Chem. **16**, 923, 1995.
36. J. Gasteiger, A. Teckentrup, L. Terfloth, and S. Spycher, J. Phys. Org. Chem. **16**, 232, 2003.
37. X. Zheng, L.H. Hu, X.J. Wang, and G.H. Chen, Chem. Phys. Lett. **390**, 186, 2004.
38. M. Sugawara, Comp. Phys. Comm. **140**, 366, 2001.
39. L.H. Holley, and M. Karplus, Proc. Natl. Acad. Sci. USA **86**, 152, 1988.
40. J.M. Boone, V.G. Sigillito, and G.S. Shaber, Med. Phys. **17**, 234, 1990.
41. F. Scarselli, and A.C. Tsoi, Neural Networks **11**, 15, 1998.
42. J.-G. Attali, and G. Pages, Neural Networks **10**, 1069, 1997.
43. J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the theory of neural computation*, Addison Wesley, Reading, 1996.

44. R. Rojas, *Theorie der Neuronalen Netze*, Springer, Heidelberg, 1996.
45. Y. LeCun, L. Bottou, G.B. Orr, and K.R. Müller, in *Neural Networks: Tricks of the Trade*, G.B. Orr, and K.R. Müller (eds.), Springer, Heidelberg, 1998.
46. K.B. Aspeslagh, Masters Thesis, Wheaton College, Norton, MA, USA, 2000.
47. T. B. Blank, S. D. Brown, A. W. Calhoun, and D. J. Doren, J. Chem. Phys. **103**, 4129, 1995.
48. C. Munoz-Caro, and A. Nino, Computers Chem. **22**, 355, 1998.
49. S. Lorenz, A. Groß, and M. Scheffler, Chem. Phys. Lett. **395**, 210, 2004.
50. H. Gassner, M. Probst, A. Lauenstein, and K. Hermansson, J. Phys. Chem. A **102**, 4596, 1998.
51. J. Behler, S. Lorenz, and K. Reuter, J. Chem. Phys. **127**, 014705, 2007.
52. A. J. Skinner, and J. Q. Broughton, Modelling Simul. Mater. Sci. Eng. **3**, 371, 1995.
53. R. Fletcher, and C.M. Reeves, Comput. J. **7**, 149, 1964.
54. E. Polak, and G. Ribiere, Revue Francaise Informat. Recherche Operationelle **16**, 35, 1969.
55. E. Polak, *Computational Methods in Optimization*, Academic Press, New York, 1971.
56. T.H. Blank, and S.D. Brown, J. Chemometrics **8**, 391, 1994.
57. S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, Science **220**, 671, 1983.
58. J. B. Witkoskie, and D. J. Doren, J. Chem. Theory Comput. **1**, 14, 2005.
59. A. Gelb, *Applied Optimal Estimation*, MIT Press, Cambridge, MA, USA, 1974.
60. F. V. Prudente, and J. J. S. Neto, Chem. Phys. Lett. **287**, 585, 1998.
61. A.C.P. Bittencourt, F.V. Prudente, and J.D.M. Vianna, Chem. Phys. **297**, 153, 2004.
62. F. V. Prudente, P. H. Acioli, and J. J. S. Neto, J. Chem. Phys. **109**, 8801, 1998.
63. T.M. Rocha Filho, Z.T. Oliveira, Jr., L.A.C. Malbouisson, R. Gargano, and J.J. Soares Neto, Int. J. Quant. Chem. **95**, 281, 2003.
64. D. F. R. Brown, M. N. Gibbs, and D. C. Clary, J. Chem. Phys. **105**, 7597, 1996.
65. K. T. No, B. H. Chang, S. Y. Kim, M. S. Jhon, and H. A. Scheraga, Chem. Phys. Lett. **271**, 152, 1997.
66. K.-H. Cho, K.T. No, and H.A. Scheraga, J. Mol. Struct. **641**, 77, 2002.
67. S. Manzhos, and T. Carrington, Jr., J. Chem. Phys. **125**, 084109, 2006.
68. S. Manzhos, and T. Carrington, Jr., J. Chem. Phys. **125**, 194105, 2006.
69. S. Manzhos, X. Wang, R. Dawes, and T. Carrington, Jr. J. Phys. Chem. A **110**, 5295, 2006.
70. S. Manzhos, and T. Carrington, Jr., J. Chem. Phys. **127**, 014103, 2007.
71. S. Lorenz, M. Scheffler, and A. Groß, Phys. Rev. B **73**, 115431, 2006.
72. J. Behler, K. Reuter, and M. Scheffler, Phys. Rev. B **77**, 115421, 2008.
73. J. Behler, B. Delley, S. Lorenz, K. Reuter, and M.Scheffler, Phys. Rev. Lett. **94**, 36104, 2005.
74. J. Ludwig, and D.G. Vlachos, J. Chem. Phys. **127**, 154716, 2007.
75. S. Hobday, R. Smith, and J. Belbruno, Modelling Simul. Mater. Sci. Eng. **7**, 397, 1999.
76. S. Hobday, R. Smith, and J. Belbruno, Nucl. Instr. Meth. Phys. Res. B **153**, 247, 1999.
77. M. Malshe, R. Narulkar, L.M. Raff, M. Hagan, S. Bukkapatnam, and R. Komanduri, J. Chem. Phys. **129**, 044111, 2008.
78. A. Bholoa, S.D. Kenny, and R. Smith, Nucl. Instr. Meth. Phys. Res. B **255**, 1, 2007.

79. J. Behler, and M. Parrinello, Phys. Rev. Lett. **98**, 146401, 2007.
80. J. Behler, R. Martoňák D. Donadio, and M. Parrinello, phys. stat. sol. (b) **245**, 2618, 2008.
81. J. Behler, R. Martoňák D. Donadio, and M. Parrinello, Phys. Rev. Lett. **100**, 185501, 2008.
82. H. Eshet, J. Behler, and M. Parrinello, in preparation (2009).